CLAIM AMENDMENTS:

The following listing of claims will replace all prior versions and listings of the claims.

1. (Currently amended) A telephone gateway device comprising:

a first memory device external to a processor, the first memory device storing one or more telecommunication protocol templates, each of the one or more telecommunication protocol templates including a plurality of virtual machine instructions to implement a telecommunication protocol using one or more ~~and to implement a~~ finite state machines and template state data;

a second memory device internal to the processor, the second memory device to store a state table of one or more ~~data related to a first~~ finite state machines; and

[[a]] the processor having access to the first memory device, the second memory device and firmware, the firmware adapted to operate a virtual machine, and wherein the processor:

causes the virtual machine to read first virtual machine instructions of a first telecommunication protocol template from the first memory device;~~, the first virtual machine instructions selected from a first telecommunication protocol template to implement a first telecommunication protocol;~~

~~stores the first virtual machine instructions in the second memory device, wherein the second memory device contains only one first virtual machine instruction at any time;~~

~~receives first template state data to initialize the first finite state machine~~

executes the first virtual machine instructions to instantiate a first finite state machine and to store a first state table based on template state data associated with the first virtual machine instructions at the second memory device, wherein the first finite state machine and the first state table specify at least a portion of a first telecommunication protocol; and

~~executes~~ implements the first <u>telecommunication protocol</u> ~~virtual machine~~
~~instructions~~ using the first ~~template~~ state <u>table</u> ~~data~~ and the first finite state
machine.

2. (Previously presented) The telephone gateway device of claim 1, wherein the first
telecommunication protocol includes at least one of a Session Initiation Protocol (SIP), a H.323
protocol, a STUN protocol, and a Dynamic Host Configuration Protocol (DHCP).

3. (Canceled).

4. (Currently amended) The telephone gateway device of claim 1, further comprising an
<u>input/output (I/O)</u> port, and wherein the processor<u>:</u>
      receives <u>input via the I/O port;</u>
      <u>determines a transition of the first finite state machine based on the input and the</u>
           <u>first state table; and</u>
      <u>updates</u> the first ~~template~~ state <u>table based on the transition</u>~~data via the I/O port~~.

5. (Currently amended) The telephone gateway device of claim 1, wherein the first
memory <u>device</u> further includes second virtual machine instructions selected from a second
telecommunication protocol template to implement a second telecommunication protocol.

6. (Previously presented) The telephone gateway device of claim 5, wherein the second
telecommunication protocol includes at least one of a Session Initiation Protocol (SIP), a H.323
protocol, a STUN protocol, and a Dynamic Host Configuration Protocol (DHCP).

7. (Currently amended) The telephone gateway device of claim 5, wherein the second
telecommunication protocol template includes one or more second ~~template~~ finite state
machines.

8 -10. (Canceled).

11. (Previously presented) The telephone gateway device of claim 1, wherein the
processor is linked to the first memory device via a data bus.

12. (Previously presented) The telephone gateway device of claim 11, wherein the data bus is a serial bus.

13. (Previously presented) The telephone gateway device of claim 11, wherein the data bus is a parallel bus.

14. (Currently amended) A method of implementing a telecommunication protocol, the method comprising:

~~receiving, at a telephone gateway device, a call at a processor to implement a first telecommunication protocol;~~

selecting a first telecommunication protocol template <u>by a processor of a telephony gateway device</u>, wherein the first telecommunication protocol template includes one or more virtual machine instructions <u>executable by a virtual machine at the telephony gateway device</u> to implement [[the]] <u>a</u> first telecommunication protocol;

~~causing a virtual machine to read~~ <u>reading</u> a first virtual machine instruction of the first telecommunication protocol template from a first memory device<u>, wherein the first memory device is coupled to the processor via a bus</u>;

initializing a first finite state machine <u>using the first virtual machine instruction, wherein</u> ~~defined by~~ the first telecommunication protocol template <u>includes</u> ~~using~~ first template state data <u>defining</u>~~to arrive at~~ a first template state <u>of the first finite state machine</u>;

~~sending a message to a called party;~~

~~when the called party does not respond to the message:~~

> ~~executing a play fast busy tone command, and~~

> ~~entering a fast busy tone state;~~

~~when a response from the called party indicates that the called party has answered:~~

> ~~executing an init vocoder instruction,~~

> ~~executing a send instruction, and~~

> ~~entering a voice state; and~~

<u>storing the first template state at a second memory device of the telephony gateway device in a first state table, wherein the second memory device is internal to the processor;</u>

<u>receiving an input at the processor via a telephony call;</u>

<u>determining, by the virtual machine, an updated template state of the first finite state machine based on the first state table and the input; and</u>

storing an updated first ~~template~~ state table ~~data in a~~ at the second memory device,
~~wherein~~ the updated first ~~template~~ state table specifying the updated template
state~~data is determined based on the first finite state machine~~.

15. (Previously presented) The method of claim 14, wherein the first telecommunication
protocol includes one of a Session Initiation Protocol (SIP), a H.323 protocol, a STUN protocol,
and a dynamic host configuration protocol (DHCP).

16. (Canceled).

17. (Currently amended) The method of claim 14, wherein the first template state data
identifies transitions of the first finite state machine~~is received by a telecommunication protocol~~
~~engine~~.

18. (Currently amended) The method of claim 14, further comprising:
receiving a [[call]] request at the processor to implement a second telecommunication
protocol;
~~selecting a second telecommunication protocol template, wherein the second~~
~~telecommunication protocol template includes one or more virtual machine~~
~~instructions to implement the second telecommunication protocol;~~
~~causing the virtual machine to read a first~~
reading a second virtual machine instruction ~~of the second telecommunication protocol~~
~~template~~ from the first memory device;
initializing a second finite state machine using the second virtual machine
instruction~~defined by the second telecommunication protocol template using~~
~~second template state data to arrive at a second template state~~; and
storing ~~updated~~ a second template state ~~data in~~ of the second finite state machine at the
second memory device in a second state table~~, wherein the updated second~~
~~template state data is determined based on the second finite state machine~~.

19. (Previously presented) The method of claim 18, wherein the second
telecommunication protocol includes one of a Session Initiation Protocol (SIP), a H.323
protocol, a STUN protocol, and a dynamic host configuration protocol (DHCP).

20 - 21. (Canceled).

22. (Currently amended) The method of claim 18, wherein the ~~first~~ second template state ~~data~~ is determined based on the ~~second~~ first finite state machine.

23. (Currently amended) The method of claim 18, further comprising determining a transition of the first finite state machine based on the second~~wherein the second template state data is determined based on the first~~ finite state machine.

24. – 46. (Canceled).

47. (Previously presented) The telephone gateway device of claim 1, wherein the first memory device has a larger storage capacity than the second memory device.

48. (Previously presented) The telephone gateway device of claim 1, wherein the first memory device is a FLASH memory device.

49. (Previously presented) The telephone gateway device of claim 48, wherein the second memory device is a RAM memory device.

50. (Currently amended) The telephone gateway device of claim 5, wherein the processor:

causes the virtual machine to read the second virtual machine instructions <u>of the second telecommunication protocol template</u> from the first memory device in response to a request to implement the second telecommunication protocol;

<u>executes the second virtual machine instructions to instantiate a second finite state machine and to store a second state table based on template state data associated with the second virtual machine instructions at the second memory device, wherein the second finite state machine and the second state table specify at least a portion of the second telecommunication protocol</u>

~~stores the second virtual machine instructions in the second memory device;~~

~~receives second template state data related to initializing a second finite state machine related to the second telecommunication protocol;~~ and

~~executes~~ <u>implements</u> the second <u>telecommunication protocol</u>~~virtual machine instructions~~ using the second ~~template~~ state <u>table</u>~~data~~ and the second finite state machine.

51 - 55. (Canceled).

56. (Currently amended) A telephone gateway device, comprising:

a first memory device coupled to a processor by a bus, the first memory device storing
   one or more telecommunication protocol templates, each of the one or more
   telecommunication protocol templates including a plurality of virtual machine
   instructions to implement a telecommunication protocol and to implement a using
   one or more finite state machinesmachine;

a second memory device integral with the processor, the second memory device to store
   state data related to a first the one or more finite state machinemachines during
   implementation of the telecommunication protocol;

[[a]] the processor having access to the first memory device[[,]] and the second memory
   device, the processor configured to:
   read first virtual machine instructions from the first memory device, the first
      virtual machine instructions selected from a first telecommunication
      protocol template to implement a first telecommunication protocol;
   store the first virtual machine instructions in the second memory device, wherein
      the second memory device contains only one first virtual machine
      instruction at any time;
   receive first template state data to initialize specifying an initial state of the first
      finite state machine; and
   execute the first virtual machine instructions using the first template state data
      [[and]] to initialize the first finite state machine and to store an initial state
      table of the first finite state machine at the second memory device.

57. (New) The method of claim 14, further comprising, in response to the input, sending
a message to a called party, wherein the updated template state is determined based on a
response to the message.

58. (New) The method of claim 57, wherein, when no response to the message is received
from the called party, the updated template state comprises a fast busy tone state.

59. (New) The method of claim 57, wherein, when the response to the message indicates that the telephony call has been answered, the method further comprises executing an init vocoder instruction.

60. (New) The method of claim 57, wherein, when the response to the message indicates that the telephony call has been answered, the updated template state comprises a voice state.

61. (New) The method of claim 4, wherein the input is received via a call using the first telecommunication protocol.